CMSC 351: Algorithms

Fall '18

Instructors:

Clyde Kruskal, `kruskal@cs.umd.edu`, Sections 0301, 0401.
Office: AVW 3215.
Mohammad Nayeem Teli, `nayeem@cs.umd.edu`, Sections 0101, 0201.
Office: AVW 1351. Office phone: 301-405-3639.

Course Overview: This course presents an introduction to the techniques for designing efficient computer algorithms and analyzing their running times. General topics include asymptotics, solving summations and recurrences, algorithm design techniques, analysis of data structures, and introduction to NP-completeness.

Text: Thomas Cormen, Charles Leiserson, Ron Rivest, and Clifford Stein, *Introduction to Algorithms*, McGraw Hill and MIT Press. Any edition.

Prerequisites: Each student is expected to know the basic concepts of programming (e.g. loops, pointers, recursion), discrete mathematics (proof by induction, sets), simple data structures (lists, stacks, queues, trees), and calculus (logarithms, differentiation, integration).

Course Work and Exams: Course work will consist of written homework assignments, a programming assignment, two midterm exams, and a final. You may discuss homework problems and general solution strategies with classmates, but you ***must*** write up the solutions yourself.

Homework assignments will be turned in on Gradescope, `https://www.gradescope.com/`.

As a courtesy to the grader, homeworks are to be written clearly and neatly. Poorly written work will not be graded. When writing algorithms be sure not only that your solution is correct, but also that it is easy for the grader to understand why your solution is correct. Part of your grade will be based not only on correctness, but also on the simplicity, clarity, and elegance of your solutions.

Exams:

The midterm exams will be:

Tuesday, October 16, from 6:00pm–8:00pm
Tuesday, November 13, from 6:00pm–8:00pm

The final exam will be:

Friday, December 14 from 4:00pm–6:00pm

If any of these exam dates are a problem for you, get in touch with Jamie Matthews now.

Piazza: We will be using Piazza (`www.piazza.com`), a question-and-answer system designed to streamline discussion outside of the classroom. It supports LaTeX, code formatting, embedding of images, and attaching of files. It will be moderated by the instructors and TAs, but students are encouraged to answer questions.

Gradescope: We will be using Gradescope to hand in assignments, and to see grades. For those who haven't used it before, Gradescope is an online submission system for class assignments and tests and for grading them quickly. In this class, we will add you to the Gradescope roster for CMSC 351 automatically. You will upload your submissions for your assignments, and they will be graded by us via a template. Please note that each submission to Gradescope MUST contain the entire assignment, not just part of it. You may submit assignments as scans of your written work or as PDFs of typed documents. Also note that Gradescope contains a "regrade request" that allows you to directly tell us if there is an issue with how we graded a problem, and we will regrade it if necessary. Please use this feature only when you are sure there has been a mistake, or are completely unclear on why you lost points from our feedback.

ELMS: We will be using ELMS to get solutions, and to see recorded lectures.

Grading: Final grades will be based on the written assignments, one programming assignment, two midterm exams, and a final exam. The weights of these will be approximately 1% for each regular homework, 2% for the programming assignment, 3% for the NP-completeness homework, 25% for each midterm, and the remaining percentage (about 35%) for the final exam.

Disability Support Services: Any student eligible for and requesting reasonable academic accommodations due to a disability is requested to provide, to the instructor in office hours, a letter of accommodation from the Office of Disability Support Services (DSS) within the first two weeks of the semester.

Course Evaluations: The Department of Computer Science takes the student course evaluations very seriously. Evaluations will usually be open during the last few weeks of the course. Students can go to www.courseevalum.umd.edu to complete their evaluations.

Syllabus: This is the current version of the syllabus. The instructors reserve the right to change it at any time.

Topics: The following is a *tentative* list of topics and readings in *approximate* order.

1. Introduction, Ch. 1,2
2. Quadratic Sorting Algorithms, Ch. 2
3. Summations, Appendix A
4. Merge Sort, Ch. 2
5. Growth of Functions, Ch. 3
6. Recurrences (Integer Multiplication) Ch. 4
7. Heapsort, Ch. 6
8. Quicksort, Ch. 7
9. Sorting in Linear Time, Ch. 8
10. Medians and Order Statistics, Ch. 9
11. Graphs and Trees, Appendix B
12. Minimum Spanning Trees, Ch. 23
13. Dijkstra's algorithm, Ch. 24.3
14. Brief introduction to NP-completeness, Ch. 34